

[illegible]

DECLARATION OF RONALD S. SCHNELL

1

5. Early in my career, I worked at the Artificial Intelligence lab at the Massachusetts Institute of Technology. I worked with Professors Patrick Winston and Marvin Minsky, two of the “founding fathers” of Artificial Intelligence (“AI”). Dr. Winston was the founder of the lab, and Dr. Minsky succeeded him as its director. Under their tutelage, I got my start in AI, specifically in the field of robotics.

6. Later, in approximately 1986, I worked at Bell Laboratories on the UNIX Operating System, writing many thousands of lines of source code representing portions of the *core kernel* of the UNIX operating system.

7. From 1988-1990, I worked at IBM Corporation on their operating system called AIX (for the IBM 370 mainframe, as well as the IBM PS/2), functioning as a programmer, and eventually a development manager.

8. From 1992-1995, I worked at Sun Microsystems as an engineer, writing tens of thousands of lines of source code to make up the core kernel, and some user level functions of the Solaris Operating System.

9. From May 1997 until February 2002, I was the Founder, President and Chief Technology Officer of Mail Call, Inc., which provided a service to business consumers, allowing them to retrieve and manage their email over the phone. In this role, I performed project design, project management, data privacy and security, and all software programming and testing. I also negotiated contracts with all partners, including GTE, Verizon, Concentric Networks, Net2Phone, IDT, Earthlink, Casio, Knight-Ridder, T-Mobile (Voicestream) and others.

10. From 2002-2005, I was a Vice President at Equifax Corporation; one of the three credit bureaus of the United States, leading software development in their Internet Marketing division.

11. Beginning in 2005 and through 2011, I was recruited to be the person to run the organization responsible for monitoring Microsoft Corporation (*US v. Microsoft, N.Y. et al. v. Microsoft*) on behalf of the U.S. Department of Justice and the Attorneys General for 19 states. This role gave me powers akin to a Special Master of the DC District Court, assisting the Plaintiffs (the Attorneys General referenced above) in reporting regularly to Hon. Colleen Kollar-Kottelly, in the largest antitrust case in U.S. history. I was responsible for directing the day-to-day enforcement of the Final Judgments on behalf of the Plaintiffs. I hired and managed a total of 93 people in order to perform these duties. A triumvirate (known as the Members) was appointed by the Parties and then confirmed by the Court in order to decide how to best run the organization. The Members then recruited me based on my vast experience running technology corporations that were mostly working with operating systems that competed with the Microsoft Windows operating system. Protective Orders and an Implementation Order prohibit any further disclosure regarding this case.

12. In addition to the above, I have also been on staff and consulted at numerous other companies, including (but not limited to) Syracuse University, Encore Computing Corporation, Harris Computer Corporation, and ITT.

13. Through my professional and academic experience, I am qualified as an expert in the field of computer science.

II. RETENTION AND EXPERT REVIEW

14. I was retained through Berkeley Research Group (“BRG”) to serve as an expert witness for Nuance Communications, Inc. (“Nuance”) in this matter. BRG is paid at the rate of \$570 per hour for my services on this matter.

15. Specifically, I was retained to review material and source code related to certain International Business Machines Corporation (“IBM”) products and technology and opine about

their relationship to “DeepQA” with respect to technology as well as evaluate what was delivered to Nuance under the Software License Agreement between Nuance and IBM dated September 30, 2010 (“SLA”).

16. As part of my retention, I have reviewed the SLA, relevant pleadings and other court filings, as well as numerous documents produced by Nuance and IBM in this litigation. I have also attended and/or reviewed transcripts of many depositions in this matter.

17. I have also evaluated over 50 million lines of IBM source code and a large set of IBM source code control system (“SCCS”) metadata. The source code produced includes code for DeepQA, Discovery, Document Conversion, Natural Language Classifier, Natural Language Understanding, Retrieve and Rank, Watson Analytics, Watson Discovery Advisor, Watson Engagement Advisor, Watson Explorer, Watson for Drug Discovery, Watson for Oncology and Watson Knowledge Studio. IBM also provided some SCCS metadata for DeepQA, Watson Discovery Advisor, Watson Engagement Advisor, Watson for Drug Discovery and Watson for Oncology.

18. The SCCS metadata includes information about each change that was made to the source code, including the person who “checked-in” the change, the date on which each change was made, information as to which files were changed, added, or deleted, programmer notes describing the change, and the actual code change itself. Using the metadata, I am able to better understand the nature of the changes that were not delivered to Nuance (as well as changes that were), with a particular focus on whether or not they were modifications, updates, upgrades, error corrections, bug fixes, or other changes.

19. My understanding is that Nuance requested SCCS Metadata for Document Conversion, Discovery, Natural Language Classifier, Natural Language Understanding, Retrieve

and Rank, Watson Explorer and Watson Knowledge Studio, however, IBM declined this request and the Court did not order its production.

III. RESERVATION OF RIGHTS

20. My testimony is based on my review as of August 14, 2019. To the extent additional information is provided by the parties, I reserve my right to amend my testimony up to and including the day of trial.

IV. SUMMARY OF EXPERT OPINIONS

21. My opinions based on the aforementioned review are fully set forth in my expert report dated February 23, 2018, which I prepared in accordance with Rule 26(a)(2)(B) of the Federal Rules of Civil Procedure (the “**Schnell Report**”) (Trial Ex. PX 153). A true and correct copy of the Schnell Report is attached hereto as Tab 1, and incorporated by reference herein.

22. Based on my education and relevant work experience in computer science, I have set forth the following definitions for the terms modifications, updates, upgrades, error corrections and bug fixes below as they apply to source code. These definitions could include both changes to existing code or the addition of new code.

- a. *Modifications* - refers to actual changes in the source code, for any reason.
- b. *Updates* - “updates” is synonymous with the term “modifications,” except that sometimes it is used to refer to “software updates,” usually outside of the source code use case. Software updates, as a term, is understood to refer to new versions of released software.
- c. *Upgrades* - refers to specific modifications that result in improvements, as opposed to the generic term “modifications,” which could be for any purpose, including purposes that do not necessarily result in improvements.
- d. *Error Corrections* - refers to modifications that result in a specific type of improvement; one which corrects defects in the source code.
- e. *Bug Fixes* - the term “bug fixes” is synonymous with “error corrections.”

- f. *Other Changes* - refers to other general changes to the source code that would not otherwise be classified through the above.

23. For the purpose of my declaration, I will use the term “Updates” to refer collectively to the above definitions of updates, upgrades, modifications, error corrections, bug fixes and other changes.

24. As discussed fully in the Schnell Report, based on my education, professional experience, and review of case materials, I formed the below opinions:

- The commercialization work performed on the DeepQA code base by the IBM Emerging Technologies Group between August 2011-June 2012 are Updates to DeepQA.
- The Watson Division of the Software Group, which was later incorporated into the Watson Group, made improvements to the DeepQA code as “commercialized” by the Emerging Technologies Group and are Updates to DeepQA.
- Watson Engagement Advisor, Watson Discovery Advisor and Watson for Oncology were all developed from and improve upon the DeepQA source code and are therefore Updates to DeepQA.
- Natural Language Classifier contains source code that is directly derived from DeepQA and therefore Natural Language Classifier contains Updates to DeepQA.
- Compared to the Updates made by the Watson Division of the Software Group and later the IBM Watson Group, the Updates made by IBM Research Group to the DeepQA source tree provided to Nuance were far less substantial and meaningful and included minimal bug fixes and other “clean up” work. My understanding is that IBM Research Group also had fewer people working on DeepQA source code provided to Nuance after the code was “forked” and that key individuals involved

with the creation of DeepQA were transferred out of the IBM Research Group and into the IBM Software Group to commercialize DeepQA.

- The source code contained in the DomainIndependent_comp source tree, which was not provided to Nuance, contains Updates to DeepQA.
- It is my opinion that there was no sound code development rationale for “forking” the DeepQA code and therefore the likely rationale behind the DeepQA code fork was to deprive Nuance of Updates to DeepQA.
- IBM developed products that improve upon the DeepQA functionality outside of DeepQA. Specifically, those products are Document Conversion, Discovery, Natural Language Understanding, Retrieve and Rank, Tone Analyzer, Watson Explorer and Watson Knowledge Studio.

25. IBM submitted the *Expert Report of Christian B. Hicks*, dated April 24, 2018 (the “**Hicks Report**”), as a rebuttal to the Schnell Report.

26. I reviewed and analyzed the Hicks Report in detail. In my view, Mr. Hicks’s report falls back on the same set of technically inappropriate justifications for IBM’s failure to deliver source code to Nuance. I have laid out my criticisms and rebuttal of the Hicks Report extensively in the May 24, 2018 Rebuttal Expert Report of Ronald S. Schnell (“**Schnell Rebuttal Report**”) (Trial Ex. 154) and have also summarized them in Section IX below. A true and correct copy of the Schnell Rebuttal Report is attached hereto as Tab 2, and incorporated by reference herein.

27. All of the source code and SCCS metadata that I reviewed were provided by IBM on the secured computer made available for my review at the office of Quinn Emanuel, IBM’s counsel. Per the parties’ agreement, I had permission to print portions of source code and SCCS

metadata. My printouts, IBMNUANCE_SOURCECODE_000001- 544 (Trial Ex. PX 207), are attached hereto as Tab 3.

28. Throughout my testimony, I reference information I learned from the deposition testimony of IBM witnesses. The relevant pages from the cited deposition testimony are compiled and attached hereto as Tab 4.

V. BACKGROUND

A. DEEPQA AND THE PARTIES' SLA

29. As discussed extensively in paragraphs 32-39 of the Schnell Report, based on my review of the documents and testimony in this case, I understand that Nuance and IBM entered into an SLA on September 30, 2010 in order for Nuance to license from IBM an “Automatic Open-Domain Question Answering software system” in “both source and object code formats, and associated documentations, toolkits and algorithms” plus Updates for [REDACTED]. [REDACTED]. Trial Ex. JX 1 (SLA at 1; SLA §§ 2.12, 3.1), attached as Tab 5. This system was commonly referred to as DeepQA.

30. The SLA provides that IBM will deliver to Nuance a copy of the “Licensed IBM Background Software, including Tools in Source Code and Object Code form, and related Documentation and Software Materials, listed in Exhibit A.” Tab 5, JX 1, SLA § 1.1.

31. The SLA defines “Licensed IBM Background Software” as “(a) all Software that exists as of the Effective Date in all available formats (including Source Code and Object Code) that is owned by, or that has been developed or licensed by the IBM Research Group, including Tools, and that is listed on Exhibit A, including any modifications, updates, upgrades, error corrections, bug fixes, diagnostic and/or testing tools, that are JDBC compliant, and other changes, if available (‘Modifications’), and if such Modifications are not contractually prohibited under a Third Party agreement, and such Modifications are available, will be timely provided to

Nuance; and where the Modifications continue to meet the scope contemplated in Article 2.1 regarding the licensing of Deep QA under this Agreement, as of the Effective Date and thereafter [REDACTED], and additional Software as agreed by the parties, provided to Nuance by IBM under the Agreement (collectively ‘Updates’); and (b) all Software Materials for such Software.” Tab 5, JX 1, SLA at Sched. A.

32. Based on the documents and testimony reviewed in this case, my understanding is that DeepQA is a large “expert system” implemented as a question answering system (“QA”) and containing millions of lines of source code. IBM has described DeepQA as “a software system and architecture of taking questions as input and producing precise answers as output.” See Tab 4, Deposition of Eric Brown (“Brown Tr.”) at 16:22-17:6.

33. As IBM describes it, “DeepQA is a software architecture for deep content analysis and evidence-based reasoning. It represents a powerful capability that uses advanced natural language processing (NLP), information retrieval, reasoning, and machine learning.” Trial Ex. PX 1, attached as Tab 6, IBM’s Redbook “Building Cognitive Applications with IBM Watson Services: Volume 1 Getting Started” (June 2017) (hereinafter “Redbook” or “IBM’s Redbook”) at 31. Further, “[t]he DeepQA architecture views the problem of automatic question-answering as a massively parallel hypothesis generation and evaluation task. DeepQA can be viewed as a system that generates a wide range of possibilities and, for each, develops a level of confidence by gathering, analyzing, and assessing evidence that is based on available data.” *Id.*

34. IBM describes the DeepQA architecture in the figure below. See *Id.* at Fig. 3-1. As indicated in the drawing, the DeepQA architecture includes submitting a question for decomposition, subjecting the question to different hypotheses, checking various answer sources, ranking potential answers, and then, ultimately, providing an answer to the question.

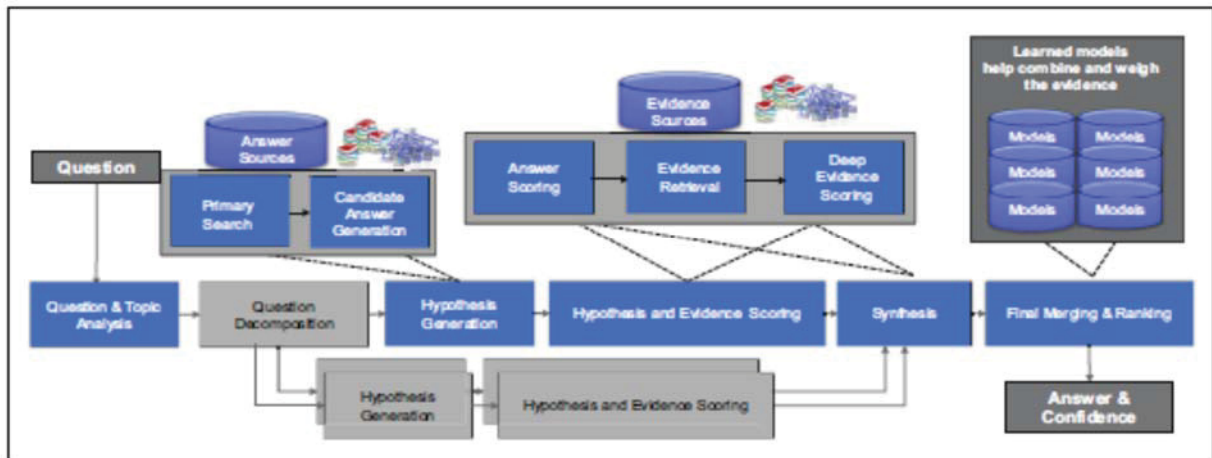


Figure 3-1 DeepQA high-level architecture

B. IBM'S "FORKING" OF THE DEEPQA SOURCE CODE AND IBM RESEARCH GROUP "FIREWALL"

35. As discussed in the Schnell Report, paragraphs 40-47, it is my opinion that IBM took elaborate steps to pass the source code originally developed by IBM Research Group to other groups within IBM and for meaningful subsequent development work to be conducted within those other groups in an attempt to avoid having to provide those Updates to Nuance.

36. Based on my review of the source code, SCCS metadata, documents, and testimony in this case, IBM forked the DeepQA source code at least twice since the execution of the SLA. The first "fork" occurred in 2011 when IBM Software Group received its own copy of the DeepQA code for commercialization and development. Trial Ex. PX 148, attached as Tab 7, IBM's Resp. to Nuance's Interrog. No. 18 ("personnel within the IBM Software Group engaged in a process to convert DeepQA into code suitable for the development of commercial products"). The second "fork" occurred in 2012 when the IBM Research Group moved its version of DeepQA into the "domainIndependent_sla" component so that it could keep the Updates it provided to Nuance separate. *See id.* at IBM's Resp. to Nuance's Interrog. No. 16 ("[P]rior to 2012, the 'domainIndependent' component contained DeepQA, including updates

and modifications IBM Research made to DeepQA. In 2012, DeepQA was moved to the ‘domainIndependent_sla’ component and, after that time, the ‘domainIndependent’ component contained other domain independent software that is not part of DeepQA.”).

37. My review of the DeepQA source code delivered to Nuance by IBM makes clear that, after IBM forked the code by providing a copy of the DeepQA source code to the IBM Software Group (“SWG”) for commercialization and further development, there was minimal to no work or resources devoted from IBM Research Group to DeepQA. For example, as IBM’s John Kelly testified at his deposition, “after the Jeopardy competition ended” IBM Research Group’s work on DeepQA was limited to “patches [and] performance improvements.” Tab 4, Deposition of John Kelly (“Kelly Tr.”) at 60:5-14.

38. Additionally, I also understand based on documents and testimony that Eric Brown, one of the key IBM Research Group developers involved with creating the DeepQA code, *see* Trial Ex. PX 139, attached as Tab 8, IBM – DeepQA Project: The Team, and other IBM Research Group personnel involved with creating DeepQA, were transferred out of the IBM Research Group and into the Watson Division and then the IBM Watson Group in order to commercialize DeepQA. For example, as IBM’s John Kelly testified at his deposition, Eric Brown was moved out of IBM Research Group because “he had the expertise to go over and help commercialize” DeepQA. Tab 4, Kelly Tr. at 47:23-49:6. IBM’s Michael Rhodin testified at his deposition that a “significant number” of IBM Research Group employees joined the Watson Group including “individuals from the Core Jeopardy! Team, the DeepQA Team.” Tab 4, Deposition of Michael Rhodin (“Rhodin Tr.”) at 20:9-21:14; *see also* Trial Ex. PX 140, attached as Tab 9, IBM J. William Murdock Profile (“I am a computer scientist in the Watson Research Center at IBM. I work for IBM Watson Group on cloud services that provide cognitive

computing capabilities to application developers. I was also a member of the original DeepQA research team when it created the original IBM Watson question answering system for the Grand Challenge task of competing with human champions in the quiz show, Jeopardy!. . . . I have worked on IBM Watson Engagement Advisor, IBM WatsonPaths, IBM Watson for Healthcare, IBM Watson Retrieve and Rank, and IBM Watson Discovery Service.”).

39. My understanding based on my review of the documents and deposition testimony in this case is that, despite IBM assuring Nuance that all work that would be done to improve DeepQA would be provided to Nuance, IBM instead erected a “logical firewall” between SWG and IBM Research Group to prevent the work being done on DeepQA outside of the IBM Research Group from being transferred to Nuance. *See* Tab 7, PX 148, IBM’s Resp. to Nuance’s Interrogatory No. 14 (“IBM Research Group employees did not have access to work product of other IBM divisions on IBM’s source code control system. Further, the managers of the IBM Research Group and the IBM Software Group were instructed that any work product of the IBM Software Group relating to DeepQA was not to be shared with the IBM Research Group without managerial approval.”).

40. Based on documents and testimony, it appears that the SWG could get updates and enhancements from IBM Research Group pertaining to DeepQA, but IBM Research Group would not receive the SWG work. Specifically, IBM’s Eric Brown testified in his deposition that “[a]nything that groups did outside of Research related to DeepQA did not flow back into the Research DeepQA.” Tab 4, Brown Tr. at 178:25-180:8. Michael Rhodin echoed that the SWG “had access to the Research DeepQA Code, but the Research Group did not have access to the Software DeepQA Code.” Tab 4, Rhodin Tr. at 71:14-72:9. IBM’s David Boloker further testified that while SWG had “access to what the Research Group was doing with the original

code base” the IBM Research Group “never” had access to what the SWG was doing with the commercial code base. Tab 4, Deposition of David Boloker (“Boloker Tr.”) at 73:21-74:4.

41. It is my understanding, based on documents and testimony, that IBM erected this “firewall” to prevent the work being done on DeepQA outside of the IBM Research Group from being transferred to Nuance. For example, IBM’s John Kelly testified at his deposition regarding the DeepQA firewall that “[t]his specific wall was put up to prevent in this case that flow through the Researchers’ work into Nuance all that investment into IBM.” Tab 4, Kelly Tr. at 189:4-190:2. IBM’s David McQueeney also testified at his deposition that it “was a deliberate choice on [IBM’s] part to not have the Research Group engage with the Software Group Team that was going to further develop that [DeepQA] code in order to respect the agreements we had made with Nuance.” Tab 4, Deposition of David McQueeney (“McQueeney Tr.”) at 54:9-55:5. These are just two examples of similar and consistent testimony regarding IBM’s explanation for its firewall that I have seen throughout IBM’s depositions in this case.

42. In my opinion, there is no sound reason as a matter of efficient code development to fork the DeepQA code in the manner IBM did. To the contrary, standard software code development is to have a common code base with all improvements flowing into the common code base. This appears to have been IBM’s understanding as well at least when it was discussing the possibility of licensing DeepQA with Nuance prior to the signing of the SLA and before it forked the code. For example, in a presentation by IBM prior to the signing of the SLA, IBM indicated that it would “Evolve the DeepQA Architecture” so that there would be “no more one-offs – looking for an extensible general purposes capability.” Trial Ex. PX 7, attached as Tab 10, Building Watson, A Brief Overview of the DeepQA Project at NUANCE00026883.

43. In fact, the “forking” of the code to keep it away from the original developers would inhibit future code development as the SWG was not able to seek input from the IBM Research Group. My opinion is that this “forking” was therefore likely done to deprive Nuance of the SWG Updates and not for any purpose related to further code development.

VI. DEEPQA SOURCE CODE SPLITS AND RESULTING UPDATES

A. IBM’S INITIAL COMMERCIALIZATION OF THE IBM RESEARCH GROUP DEEPQA CODE BASE

44. According to documents and testimony in this case, IBM forked the DeepQA code on August 1, 2011, with a full copy of the DeepQA source code being delivered to the Emerging Technologies Group (ETG) within the SWG. Trial Ex. JX 20, attached as Tab 11, IBMNUANCE00028120 (slide reflects that “Forked Research code on 8/1/11. (Periodically cherry-pick changes from Research Stream)”).

45. During the course of this litigation, Nuance requested that IBM produce all DeepQA source code, regardless of where in IBM the DeepQA code was developed.¹ IBM declined this request and the Court did not order its production.² Therefore, I have not been able to review this “commercialized” DeepQA source code base and, as such, my opinion of its contents are informed by documents and testimony in this matter.

46. As discussed fully in paragraphs 48-58 of the Schnell Report, over the course of August 2011-June 2012, David Boloker’s ETG team took what IBM acknowledged was nascent and commercially unviable DeepQA source code and made improvements to it so that it could be

¹ See Nuance Pre-Motion Letters to Judge McCarthy dated July 7, 2017 (ECF 31) and October 25, 2017 (ECF 59).

² Hearing Transcript, *Nuance Communications, Inc. v. International Business Machines Corp.*, No. 16-CV-5173 (KMK)(JCM) (Nov. 2, 2017) at 49.

turned into a commercial product. *See* Boloker Tr. at 61:2-16 (“Q. Could DeepQA have been commercialized without making these changes? A. No.”).

47. My understanding based on a review of the testimony and documents in this case is that these improvements included assessing code hygiene and then refining the source code, creating application-programming interfaces (“APIs”), building new user interfaces, removing “dead code” from the DeepQA code, as well as bug fixes that required major rewrites of the source code. *See, e.g.*, Boloker Tr. at 55:12-58:14, 59:10-60:19, 62:23-63:20, 66:16-69:18; 84:5-21; PX 43, IBMNUANCE000398895 at IBMNUANCE00039907-08 (describing “lessons learned” from Boloker’s team’s “tooling” and developments to tooling framework), attached as Tab 42; Tab 11, JX 20 at IBMNUANCE00028124 (detailing work on DeepQA “Front End”).

48. The ETG team’s work also included improvements to the tools used to build the DeepQA code as well as creating new tools for “corpus annotation” to allow for automatic, rather than manual, annotation. *See, e.g.* Tab 4, Boloker Tr. at 66:16-69:18. I understand that corpus annotation, when used in natural language processing, can assist in machine understanding of the linguistic elements of the corpus. It is extremely more efficient for this process to be automated rather than manual.

49. My understanding is that the commercialization of the DeepQA code described in paragraphs 47-48 above were necessary changes to the DeepQA code in order to turn the code into commercial products and were, in general, beneficial changes to the original DeepQA code.

50. The ETG team gave the Watson Division within the SWG this updated DeepQA code base in May and June of 2012. The SWG then began their work on this “commercialized” DeepQA source code base.

51. Despite the commercialized DeepQA source code base being developed from the original IBM Research Group DeepQA source code base, this commercialized DeepQA source code base was not provided to Nuance.

52. In my opinion, it would have taken years to develop a commercialized question-answer source code without leveraging the work done by IBM Research Group and that the ETG team benefitted from the IBM Research Group work.

53. It is my opinion, based on a reasonable degree of scientific certainty in the field of computer science, that this “commercialized” DeepQA source code base, which was not delivered to Nuance, contains Updates.

B. IBM’S WATSON DIVISION AND WATSON GROUP SUBSEQUENT DEVELOPMENT OF THE “COMMERCIALIZED” DEEPQA CODE BASE

54. Based on my review of the record in this case, my understanding is that after the ETG completed its work on the “commercialized” DeepQA source code base in the summer of 2012, that code base was provided to the Watson Division within the SWG. In January 2014, the Watson Division was then subsumed into the newly-created Watson Group.

55. As I learned from Michael Rhodin’s deposition testimony, the Watson Group then made improvements to the commercialized DeepQA code base including making it smaller, faster and more efficient, improving its accuracy and precision and creating a “Topic-Oriented Answers” pipeline. *See* Tab 4, Rhodin Tr. at 177:13-178:17.

56. These are the types of improvements that one would expect a company to make to its code in the process of turning it into a commercialized version.

57. These improvements were not provided to Nuance.

58. As noted above, though Nuance requested the production in this litigation of all DeepQA source code, regardless of where in IBM the DeepQA code was developed, it did not

receive the DeepQA source code based worked on by the SWG and/or Watson Group. I therefore have not been able to review these abovementioned improvements.

59. Based on my review, however, of other source code in evidence in this case, it is my opinion based on a reasonable degree of scientific certainty in my field of computer science, that these improvements to the “commercialized” DeepQA source code base made by the Watson Division of SWG and later the Watson Group, which were not delivered to Nuance, contain Updates.³

C. IBM RESEARCH GROUP ALSO SPLITS ITS DEEPQA CODE BASE

60. Documents and testimony in this case indicate that the IBM Research Group DeepQA source code was split in March 2012 into two different source tree components: DomainIndependent_SLA_comp (“SLAC”) and DomainIndependent_comp (“NONSLAC”). *See, e.g.* Tab 7, PX 148, IBM’s Resp. to Nuance’s Interrog. No. 16 (“[P]rior to 2012, the ‘domainIndependent’ component contained DeepQA, including updates and modifications IBM Research made to DeepQA. In 2012, DeepQA was moved to the ‘domainIndependent_sla’ component and, after that time, the ‘domainIndependent’ component contained other domain independent software that is not part of DeepQA.”).

³ In addition, during my initial review of the secured computer, IBM produced a repository entitled “LocalRepoMain.” IBM then deleted this repository from the secured computer. *See* Tab 12, Letter from Elinor Sutton to Jessica Falk dated June 21, 2017. Though I was not provided with any information as to what this repository represented, during the time it was on the computer I was able to determine that the source code was similar to the DeepQA and WEA source code that had been produced by that time. Specifically, DeepQA was referenced 232,838 times, Jeopardy was referenced 832,833 times, JClue was referenced 6,206 times and setcategory() was referenced 420 times. Setcategory() is a function used to set the category of a Jeopardy! question in DeepQA. Given these commonalities, my assumption is that this “LocalRepoMain” repository also contains Updates.

61. It is my understanding, based on testimony and evidence, that the intent of this code split was to keep track of which code should be sent to Nuance, as opposed to code that should not. *See* Trial Ex. PX 68, attached as Tab 13 (email chain discussing “Component Split To Facilitate Nuance Code Drops”). Code within the SLAC source tree was provided to Nuance while code within the NONSLAC source tree was not provided to Nuance. *See* Trial Ex. PX 70, attached as Tab 14, at IBMNUANCE00033710 (“[W]e are planning a component reorganization in RTC so that there it will be clear which code is being delivered to Nuance and which is not. . . . domainIndependent_sla_comp (Goes to nuance, licensed under the Software License Agreement [SLA]) [;] domainIndependent_comp (Does not go to nuance - we decided we don’t need to rename this component, the ‘default’ interpretation of a component name will be that it is not going to nuance’’)).

62. The SLAC source tree was provided by IBM for my review, but the NONSLAC source tree was not. IBM declined Nuance’s request for the NONSLAC source code and the Court did not order its production.⁴

63. While I did not have access to the NONSLAC source tree, evidence I reviewed from the SCCS metadata provided to me from the SLAC source tree provided some details as to what type of source code was in each source tree.

64. I found several examples of movement of code from the NONSLAC source tree to the SLAC source tree. Having reviewed the code that was moved from NONSLAC to SLAC, this code clearly contained Upgrades to the original IBM Research Group DeepQA code. By transferring this code to SLAC, it was eventually delivered to Nuance.

⁴ Hr’g Tr., *Nuance Communications, Inc. v. International Business Machines Corp.*, No. 16-CV-5173 (KMK)(JCM) (Nov. 2, 2017) at 49.

65. However, during my review, I also saw several examples of code moving from SLAC to NONSLAC. Despite the SCCS metadata indicating that this code included DeepQA Updates, because the code was moved out of the SLAC source tree into the NONSLAC source tree, these DeepQA Updates were not provided to Nuance.

66. Over the course of my review, I saw several instances of code being developed in the SLAC source tree *after* a delivery of code to Nuance, and then moved to NONSLAC *before* the next delivery to Nuance. This means that IBM Research Group developers were working on the source code within the tree provided to Nuance but then moved the fruits of their labor out of the source tree before the next delivery date so that the development work would not be provided to Nuance.

67. Some examples of topics referenced in the code going from SLAC to NONSLAC (and therefore were not provided to Nuance) include topics clearly related to DeepQA, including Q/A Pipeline, Annotations, and domain independent NLP (natural language processing).

68. In my expert report, I provide examples of IBM Research moving its code from the SLAC source code tree to the NONSLAC source tree. *See* Schnell Report ¶¶ 75-101. These paragraphs provide just some examples of source code being transferred out of SLAC to NONSLAC (and therefore, out of Nuance's reach) and are by no means exhaustive.

69. Based on my review of the source code itself and the accompanying SCCS metadata, it is clear to me that these examples of code taken out of the SLAC source code tree (which was provided to Nuance) and transferred into the NONSLAC source code tree (which was not provided to Nuance) used the same functionality as DeepQA, and that any modifications, updates, upgrades, error corrections, or bug fixes to that code would be desired for DeepQA.

70. For example, the identified transferred source code includes functionality such as “term matchers” which “compute[s] a measure of the degree of match between two terms called the term match score.” *See* Schnell Report ¶ 83 n. 50. Term matchers are used in AI technologies such as DeepQA.

71. The transferred code also provided functionality pertaining to passages scoring methods, which is another function performed by DeepQA.

72. Other transferred code included files that used a technique called Latent Semantic Analysis (LSA) in order to derive word meanings in text documents, which is also relevant to DeepQA. Based on my review of documents in this matter, IBM made a conscious decision to withhold all LSA DeepQA source code from Nuance. *See* Trial Ex. PX 63, attached as Tab 15, Email and attachments from Murthy Devarakonda to Eric Brown re: Addendum – Fw: Please review this list of proposed change sets to be accepted to Nuance delivery, dated January 5, 2012, at IBMNUANCE0031615-16 (reflecting what files would be provided in the “Nuance Devlivery (*sic*) Stream” and that the “LSA answer and scorer and server” was not included).

73. The transferred source code also included passage searching algorithms, which would also be an Update to DeepQA.

74. Other transferred source code implemented “coreference” functionality, which allows natural language parsers to find multiple words that reference the same idea, and would therefore also be a DeepQA Update.

75. Based on the above and as set out fully in Schnell Report ¶¶ 75-101, in my opinion, by transferring source code out of SLAC to NONSLAC, IBM deprived Nuance of DeepQA Updates.

76. It is my understanding that an IBM witness, Eric Brown, testified in his deposition that the NONSLAC code was not part of DeepQA and was “created by the Research Team.” Tab 4, Brown Tr. at 172:4-18. However, based on the contents of the code that was moved from NONSLAC to SLAC and vice versa, I saw no evidence that the NONSLAC code was at all different from the IBM Research Group DeepQA code that was being provided to Nuance through the SLAC source tree.

77. As indicated above, IBM refused to provide me with the NONSLAC source tree to review. Therefore, as my review was limited to only the SLAC source tree and accompanying SCCS metadata, I must assume that *all* of the NONSLAC code was originally the IBM Research Group DeepQA code. It is my opinion, based on a reasonable degree of scientific certainty in my field of computer science, that the code contained in the NONSLAC source tree, which was not delivered to Nuance, contains Updates. Further, as I did not have access to the NONSLAC source tree, I am unable to assess how IBM used the NONSLAC source tree in its product development, so there may be other Updates present in additional IBM products that were not provided to Nuance.

VII. ANALYSIS OF SPECIFIC IBM PRODUCTS’ SOURCE CODE AND ACCOMPANYING SCCS METADATA

A. OVERVIEW

78. Based on my review of source code, SCCS metadata, documents and testimony in this matter, it is my understanding that the “commercialized” DeepQA code base initially developed by the ETG and then ultimately delivered to the Watson Division and, later, the Watson group (discussed in paragraphs 44-59 above), eventually developed into the products Watson for Oncology (Oncology), Watson Engagement Advisor (WEA), and Watson Discovery Advisor (WDA).

79. My review of WEA, WDA, Oncology and DeepQA found vast overlap between these products, including that a very large majority of the domain independent code was the same. For example, I found that 71.7% of the components that were in DeepQA in September 2012 (the closest delivery prior to when the split occurred) were in WEA in 2017. My review of the Natural Language Classifier source code also uncovered overlap between it and DeepQA.

80. In paragraphs 81-115 below, and as fully set forth in Schnell Report ¶¶ 108-183, I provide examples demonstrating various overlaps, similarities and methodologies between WEA, WDA, Oncology, Natural Language Classifier and DeepQA. These examples do not represent the only files that informed my opinion and are not meant to be an exhaustive list of source code or other files that demonstrate such behavior. I reserve the right to testify regarding any of the source code files and SCCS metadata files produced by IBM as they, as a whole, informed my opinions contained in the Schnell Report.

B. WATSON ENGAGEMENT ADVISOR SOURCE CODE AND METADATA

81. My review of the Watson Engagement Advisor (WEA) source code revealed remarkable similarity and overlap between DeepQA and WEA. A complete discussion of my review of WEA and its connection to DeepQA is found at Schnell Report ¶¶ 108-142.

82. For example, my review uncovered that the word “DeepQA” appears in the WEA source code 104,562 times and the word “Jeopardy” appears in the WEA source code 406,128 times. This is indicative of a common origin between DeepQA and WEA because “DeepQA” and “Jeopardy” are not terms of art in computer programming, and the only reason to have that many references to these terms within a software product would be because of a common origin.

83. WEA also shares thousands of source code files in common with DeepQA. In the WEA “main” component alone, there are 862 files in common with DeepQA, with the same

name and author. These files originated in the IBM Research Group, have the same author, and share the same base source code.

84. Based on the programmer notes and filenames, including the main component, “qa_pipeline_comp,” the SCCS metadata for WEA also indicates that it is clearly for a Question Answering System. Moreover, the SCCS metadata for both WEA and for the SLAC source tree prior to IBM Research Group’s March 2012 forking of the DeepQA code into SLAC and NONSLAC is mostly identical.

85. As discussed in Schnell Report ¶¶ 114-118, some of the WEA SCCS programmer notes even refer to the fact that the code is based on the *Jeopardy!* supercomputer and there are references to “Jeopardy,” “JeopardyClue,” “Jeopardy Category” (or its abbreviation JCategory) and other Jeopardy-related phraseology throughout WEA.

86. The numerous references to Jeopardy in the WEA source code and SCCS metadata are further indications of the common origin between WEA and DeepQA, as my understanding is that WEA has nothing to do with the *Jeopardy!* game show.

87. There were also numerous files added to WEA along the course of its development that originated with IBM Research Group and/or the DeepQA source code. In fact, the phrase “From Research” appears in WEA metadata programmer notes one hundred and eighty-three (183) times. Specific, non-exhaustive examples of IBM Research Group code being brought into WEA are discussed in Schnell Report ¶¶ 121-130.

88. In addition to identifying the common origin and files between WEA and DeepQA, my review also uncovered entries in the WEA metadata showing DeepQA-related files that were never delivered to Nuance. Those examples are discussed in Schnell Report ¶¶ 131-133.

89. My review of the WEA source code and SCCS metadata also indicated that there are DeepQA files that were added into WEA and then updated by the IBM developers working on WEA. As discussed in Schnell Report ¶¶ 134-140, this resulted in DeepQA Updates that were not delivered to Nuance.

90. My findings based on the WEA source code and SCCS metadata are consistent with IBM internal documents and witness testimony reflecting the link between DeepQA and WEA. For example, Eric Brown testified at his deposition that “[t]o [his] knowledge, Watson Engagement Advisor and what here is called Watson Oncology Advisor . . . we now call it Watson for Oncology -- both started with DeepQA as a starting point.” Tab 4, Brown Tr. at 106:8-108:8. Rob High, IBM’s designated Rule 30(b)(6) witness on WEA, also testified that “‘Watson Engagement Advisor’” attempted to take the concept of DeepQA. . . . [and] re-used a few of the components of the original DeepQA.” Tab 4, Deposition of Rob High (“High Tr.”) at 62:14-64:15; *see also* Trial Ex. PX 162, attached as Tab 16, IBM Watson Engagement Advisor: A Cognitive Computing Linguistic Product, at 1 (“In 2011, the IBM Watson system demonstrated greater intelligence than the best human opponents in the domain of linguistically challenging factual Q&A. . . . The linguistic product that is now based on that system is called the IBM Watson Engagement Advisor.”).

91. It is my opinion, based on a reasonable degree of scientific certainty in my field of computer science, that WEA, which was not provided to Nuance, was developed from DeepQA and contains Updates.

C. WATSON DISCOVERY ADVISOR SOURCE CODE AND METADATA

92. My review of the Watson Discovery Advisor (WDA) source code revealed significant similarity and overlap between DeepQA and WDA. A complete discussion of my review of WDA and its connection to DeepQA is found at Schnell Report ¶¶ 143-158.

93. I found that WDA was very obviously similar to DeepQA with much overlap, akin to WEA. In fact, most of the files in WEA that were shared with DeepQA were also shared with WDA. The word “DeepQA” appears in WDA 91,257 times. The word “Jeopardy” appears in WDA 406,377 times. As described above, this is indicative of a common origin between DeepQA and WDA because “DeepQA” and “Jeopardy” are not terms of art in computer programming, and the only reason to have that many references to these terms within a software product would be because of a common origin.

94. WDA also shares 5,632 source code files in common with DeepQA. These files originated in the IBM Research Group and have the same author and share the same base source code.

95. There also appears to be a connection between WDA and WEA as most of the source code in WEA is also in WDA. Additionally, as shown above, some of the programmer notes in WEA refer to the fact that changes are made for the purpose of improving WDA. Outside of any user interface (UI) changes (which I did not investigate), I did not observe any core functionality differences between WEA and WDA, nor were there many file differences for the files that are common between them. Out of 364 WDA components, 301 of them were the same as WEA (which has 347 components).

96. Similarly, as with WEA, there are references to “Jeopardy,” “JeopardyClue,” “Jeopardy Category” (or its abbreviation JCategory) and other Jeopardy-related phraseology throughout WDA.

97. My review also revealed that files containing DeepQA Updates that moved from the SLAC to NONSLAC source tree prior to the Nuance deliveries ultimately became part of WDA. For example, as discussed in Schnell Report ¶¶ 93-96 and 148-149, there are term

matcher source files that were updated while in the SLAC source code tree, ultimately transferred from SLAC to NONSLAC (and therefore not delivered to Nuance), but are found in WDA.

98. My review of the WDA source code and SCCS metadata also indicated that there are DeepQA files that were added into WDA and then updated by the IBM developers working on WDA. As discussed in Schnell Report ¶¶ 150-156, this resulted in DeepQA Updates that were not delivered to Nuance.

99. My findings based on the WDA source code and SCCS metadata are consistent with IBM internal documents and witness testimony reflecting the link between DeepQA and WDA. For example, Rob High, IBM's designated Rule 30(b)(6) witness on WDA, testified that "'Watson Discovery Advisor' used a derivative of DeepQA." Tab 4, High Tr. at 74:11-76:5; *see also* Rhodin Tr. at 87:8-88:8 (noting that Watson Discovery Advisor used the DeepQA code); Trial Ex. PX 151, attached as Tab 17, IBM Watson Group, *Watson Discovery Advisor: Question-Answering in an Industrial Setting*, dated June 9, 2016 ("Open-Domain question answering received a wave of renewed attention when IBM's Watson system successfully competed with human champions in the Jeopardy! challenge. That system was developed into the Watson Discovery Advisor IBM product").

100. It is my opinion, based on a reasonable degree of scientific certainty in my field of computer science, that WDA, which was not provided to Nuance, was developed from DeepQA and contains Updates.

D. WATSON FOR ONCOLOGY METADATA AND SOURCE CODE

101. The Oncology source code that I was able to review illuminated striking similarity and overlap between DeepQA and Oncology. A complete discussion of my review of Oncology and its connection to DeepQA is found at Schnell Report ¶¶ 159-173.

102. However, as discussed in the Schnell Report ¶¶ 159-161, I did not have access to complete set of all Oncology source code or SCCS metadata, which limited my ability to fully evaluate the Oncology product. As such, there may be more similarities between Oncology and DeepQA than what I was able to uncover through my review and analysis.

103. My review of the produced Oncology source code indicates that the non-domain-specific portion of the source code uses the same code with the same filenames and authors as DeepQA. As evidence of this, Oncology references the term “DeepQA” 6,888 times in its source code. This is indicative of a common code base between Oncology and DeepQA. DeepQA itself as delivered in September 2010 only references the term “DeepQA” 4,315 times and 274 times in March 2017.

104. In addition to the commonality of terms, there were data structures shared that one would not expect for a domain-specific application such as Oncology. For example, “questions” were stored in a data structure called JClue, which stands for Jeopardy Clue. Each question was placed in a structure that had a category, question, and dollar value. This would not be what I would expect for an Oncology application, and in fact, the category and dollar value were always pre-set to a default value and were not used. Further, the directory “com.ibm.bluej” itself is used by Oncology 70,605 times – this clearly pertains to DeepQA as “BlueJ” was the original IBM codename for the Jeopardy Supercomputer.

105. My review of the Oncology source code also indicated that IBM Research Group personnel (or personnel who were formerly associated with IBM Research Group) wrote code for both DeepQA and Oncology. Such personnel include Michael S. Moore, Adam Lally and Eric Brown.

106. Based on my review, it is clear that the Oncology code is based on the DeepQA code; I discuss specific examples of DeepQA code that appears in an enhanced and improved form in Oncology in Schnell Report ¶¶ 165-168. These examples are DeepQA Updates that were not delivered to Nuance.

107. My review of the Oncology source code also revealed files that contained Updates to the DeepQA functionality, such as natural language processing and natural language annotation, that were not provided to Nuance. These functionality Updates are discussed in Schnell Report ¶¶ 169-171.

108. My findings based on the Oncology source code and SCCS metadata are consistent with IBM internal documents and witness testimony reflecting the link between DeepQA and Oncology. For example, Eric Brown testified at his deposition that “[t]o [his] knowledge, Watson Engagement Advisor and what here is called Watson Oncology Advisor -- we now call it Watson for Oncology -- both started with DeepQA as a starting point.” Tab 4, Brown Tr. at 106:8-108:8; *see also* Trial Ex. PX 150, attached as Tab 18, Business Insider Sponsor Content by IBM, *The Computing System that won ‘Jeopardy!’ is Helping Doctors Fight Cancer*, dated February 4, 2015. Rob High, IBM’s designated Rule 30(b)(6) witness on Oncology, also testified that “a portion of ‘Watson for Oncology’ uses the DeepQA architecture.” Tab 4, High Tr. at 91:1-8.

109. It is my opinion, based on a reasonable degree of scientific certainty in my field of computer science, that Oncology, which was not provided to Nuance, was developed from DeepQA and contains Updates.

E. NATURAL LANGUAGE CLASSIFIER SOURCE CODE

110. My review of the source code IBM represented to be Natural Language Classifier (“NLC”) indicated that there was also overlap between NLC and DeepQA. A complete

discussion of my review of NLC and its connection to DeepQA is found at Schnell Report ¶¶ 174-183.

111. However, as discussed in the Schnell Report ¶¶ 174-176, I did not have access to NLC SCCS metadata, which limited my ability to fully evaluate the NLC product. As such, there may be more similarities between NLC and DeepQA than what I was able to uncover through my review and analysis.

112. Based on my review of testimony in this matter, my understanding is that NLC classifies the intent of an expression, and will understand a user's question even if the same question is asked in varying ways. *See* Tab 4, Deposition of Bing Xiang ("Xiang Tr.") at 53:8-14. IBM describes NLC as an improvement to the question analysis component of DeepQA, developed in order to classify intent. Redbook at 49 (detailing the evolution of question analysis and noting that "[t]o classify intent (Figure 4-12), Watson *Natural Language Classifier* service was developed. It is a classification service, based on deep learning, that is optimized for short input text and in the order of tens to hundreds of text classes.").

113. As discussed in Schnell Report ¶¶ 179-181, my review identified numerous common files between NLC and DeepQA, including ones that had been enhanced in NLC. These enhancements are examples of Updates in NLC that would also be Updates to DeepQA but were not delivered to Nuance.

114. My findings based on the NLC source code are consistent with IBM internal documents reflecting the link between DeepQA and NLC. *See* Trial Ex. PX 107, attached as Tab 19, Rob Yates, *Introducing the IBM Watson Natural Language Classifier*, dated July 10, 2015 ("Yesterday the Watson Natural Language Classifier was released in beta as part of the Watson Developer Cloud. This is the first piece of technology that we are releasing that has come out of

the work done on the Watson Engagement Advisor and is the evolution of the technology that played Jeopardy.”). IBM’s Redbook also describes Natural Language Classifier as an evolution of the DeepQA question analysis step, noting that “[t]he question analysis step of the pipeline is now a set of highly configurable *microservices* that greatly enhance the ability to customize the system for a particular domain (Figure 4-17).” Redbook at 51; *see also* Trial Ex. PX 33, attached as Tab 20, IBM Watson Then and Now (IBM press release noting that “The original system underlying Watson, Deep QA was built on 1 API covering five technologies” and that “[t]he Watson platform and ecosystem continues to grow” and listing Natural Language Classifier as a Watson technology); Trial Ex. PX 110, attached as Tab 21, Watson: Announcing Our Largest Release of Watson Developer Cloud Services (noting that “Natural Language Classifier mimics the human ability to classify naturally expressed short phrases into categories, or classes. By applying a deep learning approach, it can classify new phrases it has not been given before.”); Trial Ex. PX 113, attached as Tab 22, Watson Question and Answer Service to be Withdrawn (noting Natural Language Classifier as one of the “services recommended for different types of question and answer capabilities,” specifically because it “allows you to [i]nterpret and classify natural language with confidence”); Trial Ex. PX 128, attached as Tab 23, Natural Language Classifier – Interpret and classify natural language with confidence (“The Natural Language Classifier service understands the intent behind text and returns a corresponding classification, complete with a confidence score. For example ‘What is the weather like today?’ or ‘Is it hot out?’ or ‘Is it going to be nice today?’ are all ways of asking about ‘temperature.’”).

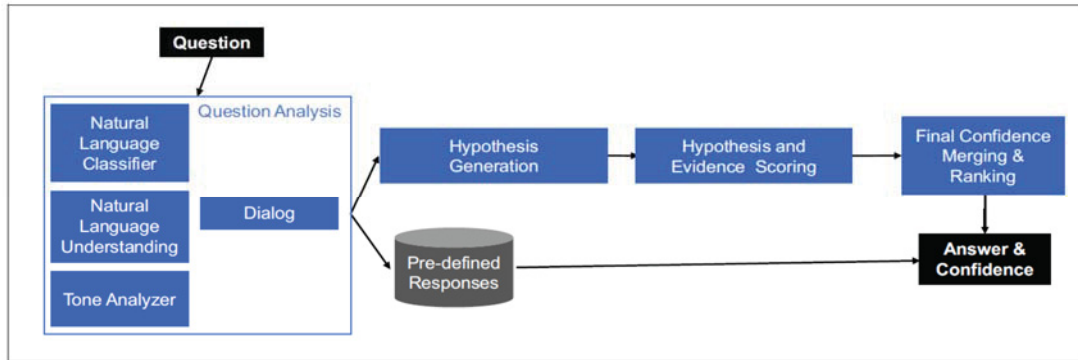


Figure 4-17 Evolution® of the DeepQA question analysis step: configurable microservices

115. It is my opinion, based on a reasonable degree of scientific certainty in my field of computer science, that NLC contains DeepQA Updates. NLC was not delivered to Nuance.

VIII. FUNCTIONAL EQUIVALENCE IN IBM WATSON PRODUCTS

116. My understanding based on my review of documents and deposition testimony in this matter is that IBM represented to Nuance that all work that would be performed to improve the functionality underlying the DeepQA pipeline, such as question analysis, hypothesis generation and scoring, and final merger and ranking, would be incorporated into DeepQA. Specifically, IBM’s Eric Brown testified at his deposition that the spirit of the SLA meant that IBM was required to provide updates if they “represented an update or upgrade to one of the existing functions of the DeepQA system,” which he described as “an open domain generic question-answering system.” Tab 4, Brown Tr. at 156:6-23, 261:9-262:19, 264:7-22. Vlad Sejnoha (Nuance’s Former Chief Technology Officer) also testified that the “spirit of the agreement was to license what’s necessary to build and evolve an open-ended question/answering system.” Deposition of Vlad Sejnoha at 31:4-9.

117. According to IBM’s Redbook, DeepQA’s functionality can be described as “Assume and pursue multiple interpretations of the question[;] Generate many plausible answers

or hypotheses[;] [and] Collect and evaluate many competing evidence paths that might support or refute those hypotheses.” Redbook at 31.

118. Nuance’s counsel asked me to consider whether certain products contained functionality that either mirrored or constituted improvements to functionality contained in DeepQA.

119. As part of this analysis, I reviewed the source code for Document Conversion, Discovery, Natural Language Understanding, Retrieve and Rank, Watson Explorer and Watson Knowledge Studio. Nuance requested the SCCS metadata for these products, however, IBM declined this request and the Court did not order its production.⁵ The inability to review SCCS metadata for these products limited my ability to opine on the connection between the products and DeepQA as I was only able to review the final version of the products’ source code and was not privy to information regarding their source code development nor programmer notes.

120. One common point between all of these products (*plus* WEA and WDA) is that they are all “RESTful” products, meaning they are able to use client-server functionality in order to offload the compute-intensive work to remote, larger and more powerful servers. IBM’s Bluemix platform is able to host these products, and using REST APIs clients can pick and choose what services they want to be performed remotely. DeepQA, as delivered to Nuance, requires that all of the computation and work be performed within a single machine. This was understandable as a requirement for playing the *Jeopardy!* game, but, in my opinion, once DeepQA was proven, that restriction became a limiting factor of the growth of the product.

⁵ Hr’g Tr., *Nuance Communications, Inc. v. International Business Machines Corp.*, No. 16-CV-5173 (KMK)(JCM) (Aug. 1, 2017) at 60.

While the other products were able to mature with these enhancements, these enhancements were kept from Nuance.

121. My conclusion based on my review of documents and testimony in this matter, is that IBM developed products that improve upon the DeepQA functionality outside of DeepQA. Specifically, those products are Document Conversion, Discovery, Natural Language Understanding, Retrieve and Rank, Tone Analyzer, Watson Explorer and Watson Knowledge Studio. My complete expert opinion on the functional equivalence of these Watson products is contained in Schnell Report ¶¶ 184-217, though I have summarized it below.

A. DOCUMENT CONVERSION

122. Based on my review of testimony in this matter, my understanding is that Document Conversion inputs data, normalizes it and allows it to be broken down into smaller pieces, which may be returned as an answer to a user's query. *See* Tab 4, Deposition of Hui Liao ("Liao Tr.") at 25:19-27:6; *see also* Trial Ex. PX 122, attached as Tab 24, Overview of the Watson™ Document Conversion Service ("The IBM Watson™ Document Conversion Service converts a single HTML, PDF, or Microsoft Word™ document. The input document is transformed into normalized HTML, plain text, or a set of JSON-formatted Answer units that can be used with other Watson services, like the Watson Retrieve and Rank Service."); Tab 21, PX 110, Watson: Announcing Our Largest Release of Watson Developer Cloud Services (noting that "Document Conversion helps you convert content in different formats (e.g., PDF) to formats that can be used by other Watson services, such as Retrieve and Rank."); Tab 22, PX 113, Watson Question and Answer Service to be Withdrawn (noting Document Conversion as one of the "services recommended for different types of question and answer capabilities," specifically because it "takes documents and 'chunks them up' into smaller answer units to return as passages"); Trial Ex. PX 174, attached as Tab 25, IBM Watson Document Conversion Service is

now available! (“The Document Conversion Service can convert all of your documents into Answer units which can then be indexed in an instance of the Retrieve and Rank Service and used by Watson to answer questions.”).

123. Specifically, my understanding is that Document Conversion will take a source document and re-organize the document to allow for easier access of information.

124. Per the Redbook, and Figure 4-22 below, Document Conversion evolved from the DeepQA pipeline and addressed problems with the Hypothesis Generation and Scoring components, which are part of DeepQA, by providing a “scalable mechanism to convert those documents into a shared format and the ability to segment those documents into relevant answer units.” Redbook at 52-53.

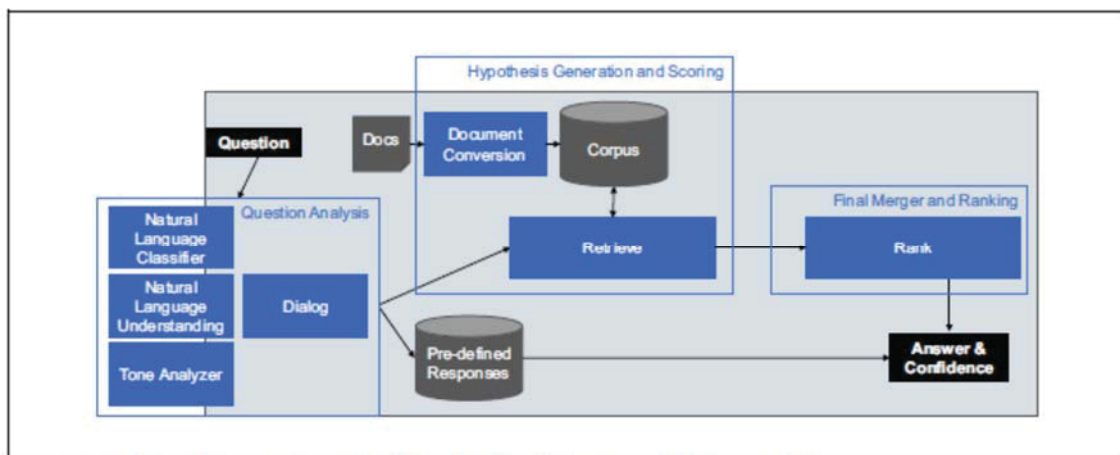


Figure 4-22 DeepQA pipeline; evolution to Watson Developer Cloud: Retrieve and Rank

125. Based on the documents and testimony provided in this matter, it is my opinion that Document Conversion improves upon the functionality underlying the DeepQA pipeline.

B. DISCOVERY

126. Based on my review of testimony in this matter, Discovery is a service that “extracts knowledge” from documents “with the intention of creating a knowledge graph,” using

natural language processing, machine learning and question analysis technology. *See* Tab 4, High Tr. at 127:12-19; Tab 4, Liao Tr. at 80:18-24, 92:11-23, 93:4-23, 119:18-22.

127. IBM’s Redbook indicates that the “Watson Discovery service provides a pipeline for ingesting, enriching and storing vast amounts of unstructured data . . . [,] allows you to run queries,” and “provides the ability to improve search results by training using documents with prior relevancy labels (relevancy ranking).” Redbook at 80.

128. Based on IBM’s description, the Discovery functionality appears to make it easier to find answers within a corpus and also improves the ranking of results. These appear to be improvements to key components of DeepQA, specifically, DeepQA’s ability to provide answers with certain confidence indications.

129. The interrelationship between DeepQA and Discovery was outlined by IBM in the diagram below. Redbook at 60, Fig. 4-31; *see also* Trial Ex. PX 119, attached as Tab 26, Watson Discovery Service: understand your data at scale with less effort (noting that IBM developed the “new Watson Discovery Service, which packages core Watson cognitive capabilities with simple tooling and APIs that enable you to quickly upload, enrich, and index large collections of your private data or public data”).

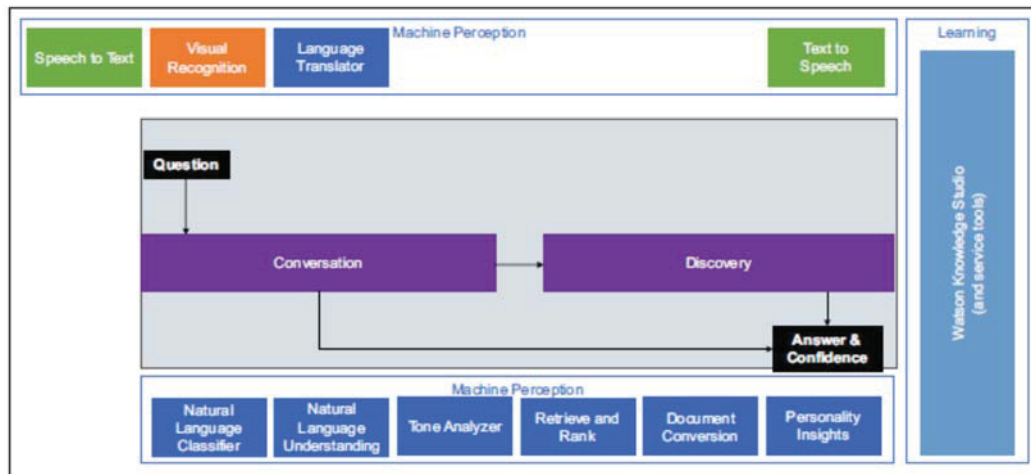


Figure 4-31 DeepQA pipeline, evolution to Watson Developer Cloud: Summary

130. Based on the documents and testimony provided in this matter, it is my opinion that Discovery improves upon the functionality underlying the DeepQA pipeline.

C. NATURAL LANGUAGE UNDERSTANDING

131. Based on my review of testimony in this matter, Natural Language Understanding (“NLU”) is a combination of three technologies used by IBM to address the problems associated with, and to expand upon, the Question Analysis component of DeepQA. *See* Tab 4, Liao Tr. at 147:19-148:23; Tab 4, High Tr. at 103:13-105:16; *see also* Tab 4, Brown Tr. at 112:24-114:4; *see also* Trial Ex. PX 124, attached as Tab 27, Top 3 reasons to move from AlchemyLanguage to Watson Natural Language Understanding (“IBM has made a number of improvements to the AlchemyLanguage API as it evolved into Watson Natural Language Understanding (NLU)”).

132. IBM has publicly stated that “the question analysis step of the [DeepQA] pipeline is now a set of highly configurable *microservices* [including NLU] that greatly enhance the ability to customize the system for a particular domain.” Redbook at 51, Fig. 4-17.

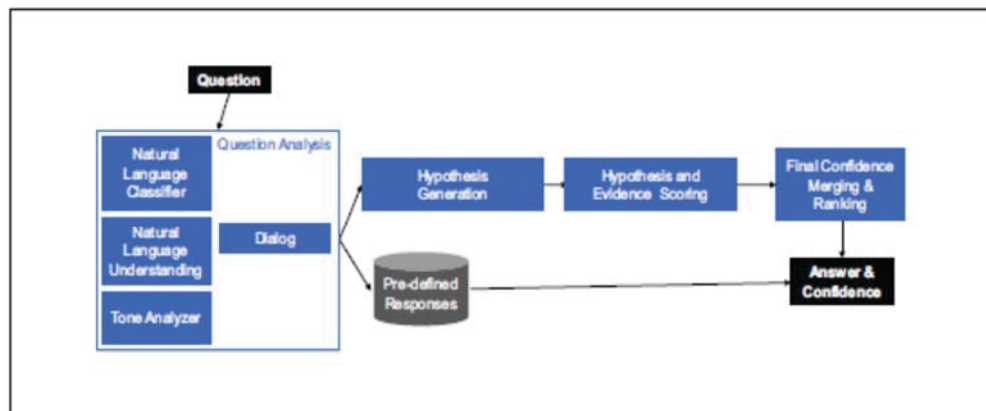


Figure 4-17 Evolution of the DeepQA question analysis step: configurable microservices

133. The NLU service’s ability to extract entities pertinent to a specific domain would be a useful addition to DeepQA as it would help DeepQA map the intent of a question to provide an appropriate response.

134. Based on the documents and testimony provided in this matter, it is my opinion that NLU improves upon the functionality underlying the DeepQA pipeline.

D. RETRIEVE AND RANK

135. Based on my review of testimony in this matter, my understanding is that Retrieve and Rank is a question and answer product reviews a vast amount of content and retrieves documents (which could be a sentence or a word) to answer natural language questions. Tab 4, Deposition of Albert Morello (“Morello Tr.”) at 36:3-9, 37:11-39:18, 66:10-14. I also understand based on testimony that Retrieve and Rank uses scoring to rank potential answer documents based on relevance. *Id.* at 49:22-52:12; High Tr. at 115:3-9.

136. IBM has publicly stated that “just like in the DeepQA pipeline, executing primary search and score and rank candidate answers is necessary Retrieve and Rank [has] a custom query builder optimized for natural language queries, a set of feature scorers to evaluate query/candidate answer overlap, and a machine learning-based ranker that can be trained with questions in the specific domain.” Redbook at 52.

137. In addition to the diagram below, in which IBM details the evolution of DeepQA to Retrieve and Rank, an IBM instructional video by a Watson Group employee explains that the Retrieve and Rank service “combines the primary search, answer scoring and ranking stages of the DeepQA pipeline into a convenient, single API pull” and that it uses “specialized information about a corpus or use case to customize the DeepQA pipeline.” Redbook at 53, Fig. 4-22 (below); *see also* Trial Ex. PX 214, attached as Tab 28, IBM Watson: DeepQA Framework and Retrieve and Rank Integration (IBM instructional video by Justin Ziniel of the Watson Group relaying that the Retrieve and Rank service “combines the primary search, answer scoring and ranking stages of the DeepQA pipeline into a convenient, single API pull” and that it uses “specialized information about a corpus or use case to customize the DeepQA pipeline.”); Tab

20, PX 33, IBM Watson Then and Now (IBM press release noting that “The original system underlying Watson, Deep QA was built on 1 API covering five technologies” and that “[t]he Watson platform and ecosystem continues to grow” and listing Retrieve and Rank as a Watson technology); Tab 21, PX 110, Watson: Announcing Our Largest Release of Watson Developer Cloud Services (noting that “Retrieve and Rank combines the power of Apache Solr with machine learning algorithms to rerank results and deliver superior answers to queries. It is especially valuable for finding information buried in vast amounts of content.”); Tab 22, PX 113, Watson Question and Answer Service to be Withdrawn (noting Retrieve and Rank as one of the “services recommended for different types of question and answer capabilities,” specifically because it “allows information retrieval with a machine learning model”); Trial Ex. PX 123, attached as Tab 29, Overview of the Retrieve and Rank service (“The purpose of the Retrieve and Rank service is to help you find documents that are more relevant than those that you might get with standard information retrieval techniques.”); Trial Ex. PX 126, attached as Tab 40, Retrieve and Rank – Building an Enhanced Information Retrieval Solution – IBM Watson Developer Cloud (“Retrieve and Rank’s ability to find the best response to a natural language query from a large set of documents is a natural fit to be used in combination with the Document Conversion service, which focuses on processing formatted documents such as PDFs, Microsoft Word documents, and HTML pages.”).

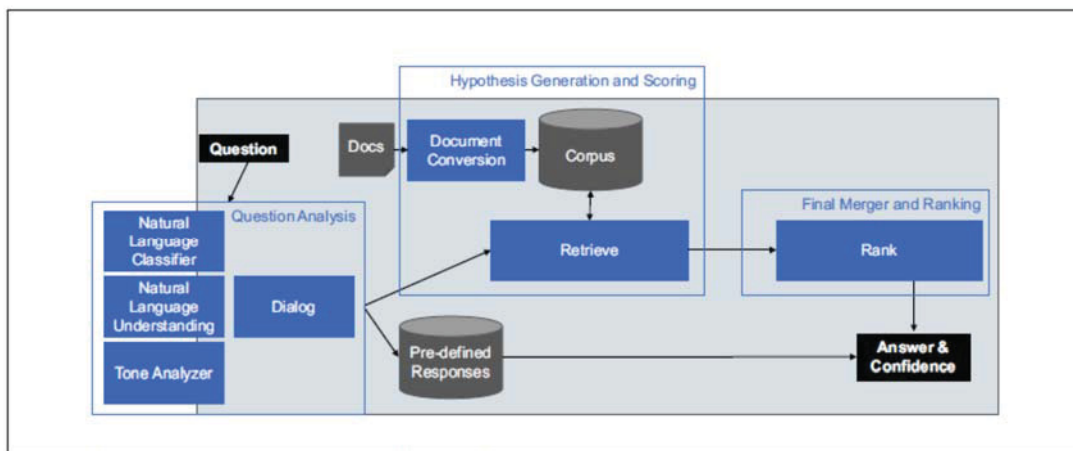


Figure 4-22 DeepQA pipeline; evolution to Watson Developer Cloud: Retrieve and Rank

138. Based on the documents and testimony provided in this matter, it is my opinion that Retrieve and Rank improves upon the functionality underlying the DeepQA pipeline.

E. TONE ANALYZER

139. According to IBM’s Redbook, the Watson Tone Analyzer service “uses linguistic analysis to detect three types of tones in text: emotions, social tendencies, and writing style” so that it may “understand emotional context of conversations and communications in order to respond in an appropriate manner.” Redbook at 49.

140. Per the graphic below from IBM’s Redbook, like NLC and NLU, Tone Analyzer improves upon the Question Analysis component of DeepQA. Redbook at 51, Fig. 4-17.; *see also* Tab 20, PX 33, IBM Watson Then and Now (IBM press release noting that “The original system underlying Watson, Deep QA was built on 1 API covering five technologies” and that “[t]he Watson platform and ecosystem continues to grow” and listing Tone Analyzer as a Watson technology).

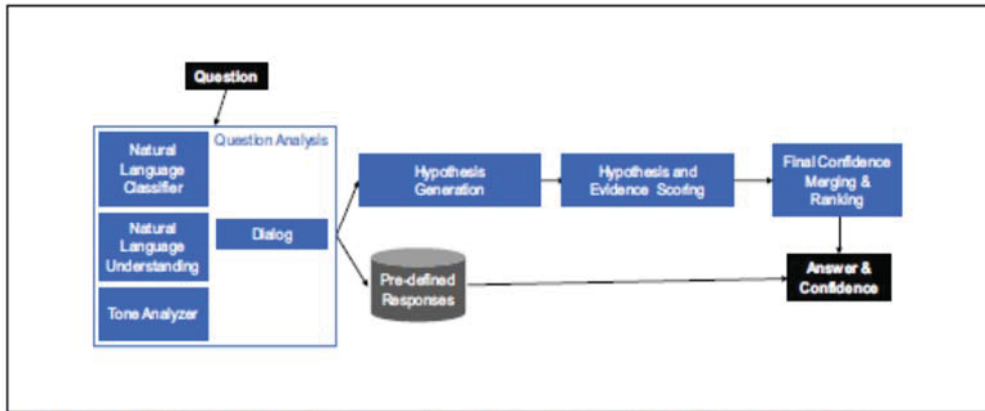


Figure 4-17 Evolution of the DeepQA question analysis step: configurable microservices

141. Tone Analyzer’s ability to ascertain the user’s emotional context also would help DeepQA identify the user’s intent and provide an appropriate response to a user’s query.

142. Based on the documents and testimony provided in this matter, it is my opinion that Tone Analyzer improves upon the functionality underlying the DeepQA pipeline.

F. WATSON EXPLORER

143. Based on my review of testimony in this matter, my understanding is that the Watson Explorer product is an information retrieval system that returns documents in response to a user’s natural language query or expression of interest in certain kinds of information. *See* Tab 4, Liao Tr. at 208:17-22, 210:12-21; *see also* Trial Ex. PX 131, attached as Tab 30, Watson Explorer (“Watson Explorer is a cognitive search and content analysis platform that gives you access to [i]nsights from all the data you care about . . . [s]earch and analyze structured, unstructured, internal, external and public content to uncover trends and patterns that improve decision-making, customer service and ROI.”).

144. IBM has publicly advertised that users can access DeepQA through Watson Explorer. *See* Trial Ex. PX 118, attached as Tab 31, Hackathon: Build with Watson™ on IBM Bluemix™ – CogniHack Lagos 2016, at 9, 14 (hereinafter “CogniHack”) (indicating “How does

Watson work? DeepQA Architecture” and “How do you access Watson? . . . Watson Explorer”).

145. IBM witnesses and documents also indicated that Watson Explorer was based in part on Watson Content Analytics, which IBM advertised as “power[ing] Watson to a *Jeopardy!* win.” See Trial Ex. PX 38, attached as Tab 32, IBM, *IBM Content Analytics Software Powers Watson to a Jeopardy! Win*, dated April 2011; see also Trial Ex. PX 133, attached as Tab 33, Overview – Watson Content Analytics (“IBM® Watson Content Analytics collects and analyzes structured and unstructured content in documents, email, databases, websites, and other enterprise repositories.”).

146. My understanding based on documents and testimony is also that Alfio Gliozzo, one of the architects of DeepQA, assisted the Watson Explorer team in advancements in “natural language query capability,” specifically, “how to recognize acronyms.” See Tab 4, Liao Tr. at 182:9-183:17; Trial Ex. PX 135, attached as Tab 34, IBM Research, *The DeepQA Research Team*.

147. Based on the documents and testimony provided in this matter, it is my opinion that Watson Explorer improves upon the functionality underlying the DeepQA pipeline.

G. WATSON KNOWLEDGE STUDIO

148. Based on my review of documents in this matter, my understanding is that “[t]he primary purpose of Watson Knowledge Studio is to help create a model that understands domain specific linguistic nuances, meanings, and relations,” and “provides a rule-based model to find entities in documents” in order to “help Watson become a subject matter expert in a given domain or industry.” Redbook at 82; see also Trial Ex. PX 129, attached as Tab 35, Watson Knowledge Studio – Overview – IBM Watson Developer Cloud (“Use IBM Watson™ Knowledge Studio to create a machine-learning model that understands the linguistic nuances,

meaning, and relationships specific to your industry or to create a rule-based model that finds entities in documents based on rules that you define.”).

149. The functionality of Watson Knowledge Studio would therefore be beneficial to training DeepQA to be used in different domains or industries outside of the *Jeopardy!* context.

150. Per IBM’s Redbook, Watson Knowledge Studio “provides the tools to teach Watson the unique characteristics of your domain” and has “fingerprints” of DeepQA. Redbook at 60, Fig. 4-31.

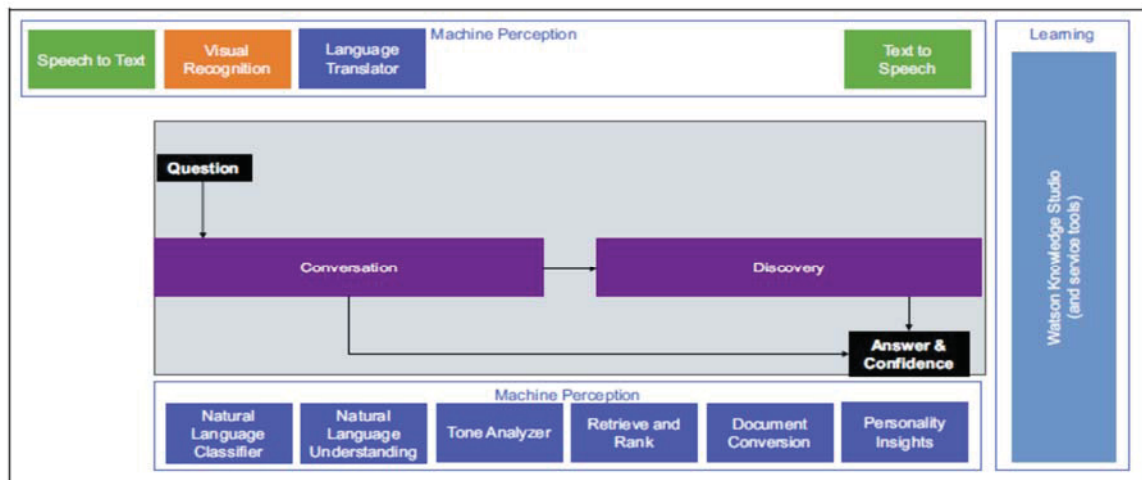


Figure 4-31 DeepQA pipeline, evolution to Watson Developer Cloud: Summary

151. IBM has also publicly advertised that Watson Knowledge Studio began with a study involving the IBM Research Group and that users can access DeepQA through Watson Knowledge Studio. See Tab 31, PX 118, CogniHack at 9, 14 (indicating “How does Watson work? DeepQA Architecture” and “How do you access Watson? . . . Watson Knowledge Studio”); see also Trial Ex. PX 117, attached as Tab 36, IBM Watson Knowledge Studio – Teach Watson about your domain (“The effort behind IBM Watson Knowledge Studio began almost two years ago when a joint team from IBM Research and the IBM Watson Group did a

study to understand existing natural language processing (NLP) technologies and best practices.”).

152. In my review of the DeepQA source code on the secured computer, I also noted references to Watson Knowledge Studio.

153. Based on the documents and testimony provided in this matter, it is my opinion that Watson Knowledge Studio improves upon the functionality underlying the DeepQA pipeline.

IX. RESPONSE TO HICKS REPORT

A. GENERAL REBUTTAL

154. As discussed in paragraphs 25-26 above, I submitted the Schnell Rebuttal Report in response to the Hicks Report. While my rebuttal report addresses the Hicks Report at length, I have highlighted some of my responses to the Hicks Report below.

155. As discussed in detail in the Schnell Rebuttal Report, the Hicks Report uses the same technically inappropriate justifications for IBM’s failure to deliver DeepQA Updates to Nuance. *See* Schnell Rebuttal Report ¶¶ 5-12.

156. Mr. Hicks opines that IBM was only obligated to deliver Updates to files specifically listed in Exhibit A of the SLA. However, as I discuss in the Schnell Rebuttal Report ¶¶ 6-7 and 18-20, it is not technically sound to limit Updates to a software project to those that update a specific set of existing files as this would greatly hinder a programmer’s ability to make Updates to the software. The term “updates” has a significantly broader definition in the software development industry and includes new files that would improve the functionality licensed. It is, in fact, standard industry practice to add files while updating software. Moreover, IBM’s initial delivery of DeepQA to Nuance in September 2010 included many files that were not listed in Exhibit A, and IBM continued to deliver files not in Exhibit A after the initial

delivery. As of my last review of the secured computer, 2,062 .java files were delivered to Nuance that were not in Exhibit A to the SLA.

157. Mr. Hicks's opinion also relies on IBM's interpretation of the SLA, that is, Mr. Hicks has assumed, without analyzing, that any and all work performed on DeepQA outside of the IBM Research Group are outside the scope of the SLA and therefore are not Updates. As Mr. Hicks did not offer any technical basis for this opinion, I do not offer any technical rebuttal.

158. Mr. Hicks further attempted to distinguish the use of DeepQA in IBM products from DeepQA Updates. As discussed in Schnell Rebuttal Report ¶¶ 9-11, this ignores the fact that the DeepQA source code used in the IBM products are central to the operation of those products and an admission that IBM developed Updates that were not delivered to Nuance. It is also standard practice in the industry to have a common core across products that share core functionality. It is my opinion, based on a reasonable degree of scientific certainty in my field of computer science, that whether or not the products used the DeepQA source code as their core or were themselves a modified version of the DeepQA core does not change the fact that they contained DeepQA Updates.

B. SPECIFIC REBUTTALS

1. MR. HICKS'S MISCHARACTERIZATION OF MY DEFINITIONS

159. In his report, Mr. Hicks took the terms I defined in paragraph 22 of this testimony out of context. As I discuss in Schnell Rebuttal Report ¶¶ 13-20, I have set forth definitions for those terms in the industry specifically as they relate to source code. Mr. Hicks, instead, attempts to argue that all of the definitions pertain to "a set of existing source code" and therefore, the DeepQA Updates would only be to those files listed on Exhibit A. For the reasons noted above, I disagree with Mr. Hicks.

160. Moreover, limiting Updates to “existing source code” only would defy even a common sense understanding of source code development as it would be impossible to run a software project without the addition of software code that previously did not exist. Best coding practice is to make code as “modular” as possible, therefore, software developers add files they add functionality. My review of the source code in this matter revealed that the IBM programmers followed this best practice and created modular code, which necessitated the addition of files for Updates outside of Exhibit A.

2. *IBM’S FORKING OF THE DEEPQA CODE*

161. In his report, Mr. Hicks states that I am not qualified to render an opinion as to IBM’s reasons for forking the source code. As I indicate in Schnell Rebuttal Report ¶¶ 21-25, I have not offered an opinion as to IBM’s motivation. As discussed in paragraphs 35-43 above, as IBM admits in the documents and testimony provided in this case, IBM forked the DeepQA code to keep Nuance from receiving work product from outside of the IBM Research Group. My expert opinion is limited to opining that no sound code development rationale exists for the DeepQA “forks.”

3. *DISCUSSION OF SLAC AND NONSLAC RTC COMPONENTS*

162. As discussed in paragraphs 17-18 above, as part of my analysis, I reviewed SCCS metadata on the assorted IBM products. This SCCS metadata for these products was maintained in Rational Team Concert (RTC), a software development tool.

163. In his report, Mr. Hicks uses a limited definition of RTC components. Specifically, that RTC components are organizational units for source code within a single codebase. While that is one way that RTC components can be used, it is not the only way.

164. Mr. Hicks uses this limited definition to assert that the SLAC and NONSLAC components I discuss in paragraphs 60-77 above were just reorganized source code and that the IBM Research Group did not make any additional copies of the DeepQA source code.

165. As discussed in Schnell Rebuttal Report ¶¶ 26-28, my review revealed that there were files in the SLAC and NONSLAC components with different content in each component. It was clear from the architecture of the code and from the programmer notes that the NONSLAC components made use of files in the SLAC component, but not the other way around. Further, Mr. Hick's assertion that the NONSLAC code was essentially no different than the SLAC code does not comport with IBM's refusal to produce the NONSLAC code to Nuance.

166. Mr. Hicks also takes issue with my adoption of the SLAC and NONSLAC nomenclature. However, as discussed in Schnell Rebuttal Report ¶¶ 29-31, I have merely adopted IBM's terminology.

167. As IBM refused to provide access to the NONSLAC source code, I must assume that the NONSLAC source code was created from an additional copy of the original IBM Research Group DeepQA code.

4. *DISCUSSION OF OVERLAP IN IBM PRODUCTS*

168. I have numerous critiques of the Hicks Report and its analysis of the overlap in specific IBM products. These critiques are discussed at length in Schnell Rebuttal Report ¶¶ 32-79. I have briefly touched on each criticism below.

a. *WATSON ENGAGEMENT ADVISOR*

169. My report analyzed the percentage of DeepQA present in WEA, however, the Hicks Report compared the inverse, that is, the percentage of WEA filenames present in DeepQA. It therefore is not an appropriate comparison.

170. Further, Mr. Hicks's assessment of the number of files within WEA is not based on a proper methodology as it includes all files (not just source code files) and therefore overstates the number of WEA files. My percentage analysis was also limited to the two components relevant to question and answer functionality whereas Mr. Hicks reviewed all components. Therefore, the components I left out may contain additional file names that match DeepQA. Adding these components when calculating percentage of DeepQA in WEA would not have made the percentage any lower, as I included all of the obvious source code from DeepQA.

171. Using Mr. Hicks's preference for counting filenames instead of components, I found that of the 5,506 filenames in the September 2010 delivery of DeepQA that were obviously source code, 3,022 were present in WEA (54.89%). Of the 5,070 filenames in the September 2012 delivery of DeepQA that were obviously source code, 2,920 were present in WEA (57.59%). In both cases, a majority of DeepQA filenames were present in WEA.

172. Mr. Hicks also stated that some WEA modifications were insignificant. My understanding is that this opinion has no relevance as the SLA does not limit Nuance to significant Updates. Therefore, even if the Updates were insignificant, they should have been provided to Nuance.

b. *WATSON DISCOVERY ADVISOR*

173. My report analyzed the percentage of DeepQA present in WDA, however, the Hicks Report again compared the inverse, that is, the percentage of WDA filenames present in DeepQA. It therefore is not an appropriate comparison.

174. Further, Mr. Hicks's assessment of the number of files within WDA is not based on a proper methodology as it includes all files (not just source code files) and therefore overstates the number of WDA files. My percentage analysis was also limited to the two

components relevant to question and answer functionality whereas Mr. Hicks reviewed all components. Therefore, the components I left out may contain additional file names that match DeepQA. Adding these components when calculating percentage of DeepQA in WDA would not have made the percentage any lower, as I included all of the obvious source code from DeepQA.

175. Again, using Mr. Hicks's preference for counting filenames instead of components, for the September 2010 DeepQA delivery, out of the 5,506 files that I considered to be obviously source code, I found that 3,991 of those files were in WDA (72.48%).

c. *WATSON ONCOLOGY*

176. As discussed extensively in Schnell Rebuttal Report ¶¶ 44-50, it appears that Mr. Hicks was provided access to certain Watson Oncology files that I was not. Therefore, while I am unable to conduct an apples-to-apples comparison to Mr. Hicks's discussion of Watson Oncology, I was able to uncover 554 obvious source code filenames from DeepQA that were in source code form in what was produced as the Oncology source code. Further, because the DeepQA core is used in a similar way in WEA and WDA, it is logical to infer that Oncology also uses thousands of DeepQA files.

d. *FUNCTIONAL EQUIVALENCE IN IBM PRODUCTS*

177. As discussed in Schnell Rebuttal Report ¶¶ 51-56, the Hicks Report does not dispute the characterizations of DeepQA's functionality nor of the functionality of the other IBM products that I discuss.

5. *DISCUSSION OF SPECIFIC SOURCE CODE FILES*

178. Mr. Hicks purports to address specific files that I have identified as showing common origin or overlap with DeepQA. I have addressed my criticisms of Mr. Hicks's

analysis of these specific files at Schnell Rebuttal Report ¶¶ 57-79 and briefly discuss them below.

179. Broadly speaking, Mr. Hicks makes no technical assessment of whether the files themselves improved upon the DeepQA source code or functionality and makes no effort to opine that the identified files are not relevant to DeepQA. At best, Mr. Hicks details that some of the specific files did not appear to have “significant” changes, but, as noted previously, my understanding is that the SLA did not limit Nuance to “significant” updates. Instead, Mr. Hicks either falls back on his assumptions regarding IBM’s interpretation of the SLA to conclude that Nuance was not entitled to these files or blindly accepts references in the SCCS metadata that certain files were inadvertently included.

180. As previously discussed, given IBM’s refusal to provide the NONSLAC source code, I was hindered in my ability to fully review and compare the DeepQA source code. To the extent that Mr. Hicks was able to review the NONSLAC source code, I reserve my right to update and/or amend my testimony.

X. IBM’S REFUSAL TO COOPERATE FULLY IN THE DISCOVERY PROCESS

181. As indicated throughout this declaration, Nuance requested certain source code and SCCS metadata that IBM refused to produce. The Court also did not order its production. My view is that IBM’s refusal to produce the i) commercialized DeepQA code, ii) the NONSLAC source code tree, and iii) SCCS metadata for Document Conversion, Discovery, Natural Language Classifier, Natural Language Understanding, Retrieve and Rank, Watson Explorer and Watson Knowledge Studio all impacted my ability to further assess the extent to which IBM developed DeepQA Updates that it withheld from Nuance.

182. My understanding is that, in addition to refusing to produce certain requested discovery, IBM also was not forthcoming in responding to Nuance’s requests for “all documents

and communications regarding updates to the DeepQA software.” For example, though IBM agreed to produce “documents and communications regarding the creation of, division of work for, distribution of work for, progress of, and/or implementation of updates to the DeepQA software,” IBM did not produce any documents regarding Watson Discovery Advisor (WDA) until after Nuance expressly identified the product. *See* Trial Ex. PX 138, July 10, 2017 Letter from Jessica Falk to Kevin Reed and Elinor Sutton, attached as Tab 41.

183. I also understand that IBM expressly refused to answer Nuance’s interrogatories focused on identifying any additional IBM products or development work utilizing question answering technology. *See* Tab 7, PX 148, IBM’s Objections and Responses to Nuance’s Third Set of Interrogatories. Therefore, Nuance does not know whether there are other products developed by IBM that utilize question answer technology that have been kept from Nuance.

184. Further, after fact discovery in this matter closed, IBM introduced two new products that appear based on DeepQA technology. IBM introduced Project Debater, an AI question answering technology that “built on” the Watson *Jeopardy!* foundation. *See* Trial Ex. PX 144, attached as Tab 37, AI Learns the Art of Debate (Project Debater “is the latest in a long line of major AI innovations at IBM, which also include . . . IBM Watson, which beat the top human champions on Jeopardy! in 2011”). IBM also introduced Watson Assistant, which it advertises as “know[ing] when to provide a direct answer to a common question or reference more generalized search results for something more complex.” Trial Ex. PX 194, attached as Tab 38, IBM Cloud Docs, Watson Assistant, Skills; *see also* Trial Ex. PX 145, attached as Tab 39, IBM Watson Assistant gets smarter and faster, making customer service a breeze. Despite both products clearly containing question answering technology, my understanding is that IBM did not provide any documents or testimony in this matter pertaining to Project Debater or

Watson Assistant. I therefore did not have the opportunity to examine either product in the course of my expert review.

185. Based on the above, to provide a proper assessment of which products contain DeepQA Updates, a full audit of IBM is warranted.

XI. CONCLUSION

186. As indicated above, my review of documents, deposition testimony, IBM source code and SCCS metadata leads me to conclude that, after IBM “forked” the DeepQA code to provide a copy of the IBM Research Group DeepQA code to the Emerging Technologies Group (which then relayed their version of the code to the Watson Division of the IBM Software Group and ultimately, the Watson Group), IBM made numerous Updates to DeepQA outside of the IBM Research Group. These Updates were not provided to Nuance. These Updates include, but are not limited to, the “commercialized” DeepQA code base as well as the products Watson Engagement Advisor, Watson Discovery Advisor, Watson for Oncology, and Natural Language Classifier.

187. My review of documents, deposition testimony, IBM source code and SCCS metadata also lead me to conclude that the IBM Research Group also “forked” its version of the DeepQA code base into two different source trees *within* the IBM Research Group: DomainIndependent_SLA_comp (“SLAC”) and DomainIndependent_comp (“NONSLAC”), of which only the SLAC source tree was provided to Nuance. Based on my review, I conclude that the NONSLAC source tree contains Updates to DeepQA.

188. There was no sound code development rationale for “forking” the DeepQA code. Accordingly, my opinion is that the likely basis for “forking” the code was to deprive Nuance of Updates to DeepQA. This understanding is supported by IBM documents and testimony that I

have reviewed in this matter, which indicate that IBM sought to keep Updates from being provided to Nuance.

189. To the extent there were IBM Research Group Updates provided to Nuance, those enhancements were actually copied over to other IBM products as well.

190. In my opinion, compared to the Updates made by the rest of IBM, the Updates delivered by the IBM Research Group to Nuance were minimal.

191. Finally, based on my review of documents and testimony in this matter, my opinion is that the products Document Conversion, Discovery, Natural Language Understanding, Retrieve and Rank, Tone Analyzer, Watson Explorer and Watson Knowledge Studio all improve upon the functionality of DeepQA.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Ronald S. Schnell', with a stylized, sweeping flourish at the end.

Ronald S. Schnell

August 14, 2019